call void @llvm.dbg.value(metadata ptr %a, metadata !24, metadata ...!DIExpression()),!dbg!31 call void @llvm.dbg.value(metadata ptr %b, metadata !25, metadata ...!DIExpression()), !dbg!31 call void @llvm.dbg.value(metadata ptr %c, metadata !26, metadata ...!DIExpression()),!dbq!31 call void @llvm.dbg.value(metadata ptr %cond, metadata !27, metadata ... !DIExpression()), !dbg !31 call void @llvm.dbg.value(metadata i32 %n, metadata !28, metadata ... !DIExpression()), !dbg !31 tail call void asm sideeffect ".inst 0x2520e020", ""() #7, !dbg !32, !srcloc call void @llvm.dbg.value(metadata i32 0, metadata !29, metadata ...!DIExpression()),!dbg!35 %cmp51 = icmp sgt i32 %n, 0, !dbg !36br i1 %cmp51, label %for.body.preheader, label %for.cond.cleanup, !dbg !38 F for.body.preheader: %wide.trip.count = zext i32 %n to i64, !dbg !36 %0 = tail call i64 @llvm.vscale.i64(), !dbg !38 %1 = shl i64 %0, 1, !dbg !38 %2 = shl i64 %0, 2, !dbg !38 %.not = icmp ugt i64 %2, %wide.trip.count, !dbg !38 br i1 %.not, label %for.body.preheader12, label %pre.alc, !dbg !38 pre.alc: %7 = tail call < vscale x 2 x i1 > @llvm.aarch64.sve.ptrue.nxv2i1(i32 31)%8 = tail call <vscale x 2 x i64> @llvm.aarch64.sve.index.nxv2i64(i64 0, i64 %9 = urem i64 %wide.trip.count, %1 %total.iterations.to.be.vectorized = sub nsw i64 %wide.trip.count, %9 %10 = load < vscale x 2 x i8 >, ptr %cond, align 2 %11 = icmp ne <vscale x 2 x i8> %10, zeroinitializer %12 = tail call i64 @llvm.aarch64.sve.cntp.nxv2i1(<vscale x 2 x i1> %11, ... <vscale x 2 x i1> %11) br label %alc.header alc.header: %index = phi i64 [ %1, %pre.alc ], [ %67, %new.latch ] %uniform.vector = phi <vscale x 2 x i64> [ %8, %pre.alc ], [ %64, %new.latch %uniform.vector.predicates = phi <vscale x 2 x i1> [ %11, %pre.alc ], [ %65, .. %new.latch ] %uniform.vec.actives = phi i64 [ %12, %pre.alc ], [ %66, %new.latch ] %13 = getelementptr inbounds i8, ptr %cond, i64 %index, !dbg !43 %14 = load <vscale x 2 x i8>, ptr %13, align 2 %15 = icmp ne <vscale x 2 x i8> %14, zeroinitializer %16 = tail call i64 @llvm.aarch64.sve.cntp.nxv2i1(<vscale x 2 x i1> %15, .. <vscale x 2 x i1> %15) %17 = add i64 %16, %uniform.vec.actives %condition.not = icmp ugt i64 %17, %1 br i1 %condition.not, label %linearized, label %lane.gather lane.gather: %18 = tail call <vscale x 2 x i64> @llvm.aarch64.sve.index.nxv2i64(i64 ... %index, i64 1) %19 = tail call <vscale x 2 x i64> @llvm.aarch64.sve.compact.nxv2i64(<vscale ... x 2 x i1> %uniform.vector.predicates, <vscale x 2 x i64> %uniform.vector) %20 = tail call <vscale x 2 x i64> @llvm.aarch64.sve.compact.nxv2i64(<vscale ... x 2 x i1> %15, <vscale x 2 x i64> %18) %21 = tail call <vscale x 2 x i1> @llvm.aarch64.sve.whilelt.nxv2i1.i64(i64 ... 0, i64 %uniform.vec.actives) %22 = tail call <vscale x 2 x i64> @llvm.aarch64.sve.splice.nxv2i64(<vscale ... x 2 x i1> %21, <vscale x 2 x i64> %19, <vscale x 2 x i64> %20) %23 = tail call <vscale x 2 x i1> @llvm.aarch64.sve.whilelt.nxv2i1.i64(i64 ... 0, i64 %17) %24 = tail call i64 @llvm.aarch64.sve.cntp.nxv2i1(<vscale x 2 x i1> %23, ... <vscale x 2 x i1> %23) %25 = icmp eq i64 %17, %1br i1 %25, label %uniform.block, label %new.latch F uniform.block: %26 = tail call < vscale x 2 x i32 >... @llvm.aarch64.sve.ld1.gather.index.nxv2i32(<vscale x 2 x i1> %7, ptr %a, <vscale x 2 x i64> %22) %27 = mul < vscale x 2 x i32 > %26, shufflevector (< vscale x 2 x i32 > %26) linearized: %46 = getelementptr inbounds i32, ptr %a, i64 %index, !dbg !52 ... insertelement (<vscale x 2 x i32> poison, i32 18, i64 0), <vscale x 2 x i32> %47 = getelementptr inbounds i32, ptr %c, i64 %index, !dbg !57 ... poison, <vscale x 2 x i32> zeroinitializer) %48 = getelementptr inbounds i32, ptr %b, i64 %index, !dbg !60 %28 = tail call < vscale x 2 x i32 >%49 = tail call <vscale x 2 x i32> @llvm.masked.load.nxv2i32.p0(ptr %46, i32 ... @llvm.aarch64.sve.ld1.gather.index.nxv2i32(<vscale x 2 x i1> %7, ptr %c, ... 8, <vscale x 2 x i1> %15, <vscale x 2 x i32> undef) ... <vscale x 2 x i64> %22) %50 = mul <vscale x 2 x i32> %49, shufflevector (<vscale x 2 x i32>  $%29 = \text{shl} < \text{vscale } x \ 2 \ x \ i32 > \%28$ , shufflevector ( $< \text{vscale } x \ 2 \ x \ i32 > \%28$ ) ... insertelement ( $\langle vscale \times 2 \times i32 \rangle$  poison, i32 18, i64 0),  $\langle vscale \times 2 \times i32 \rangle$ ... insertelement (<vscale x 2 x i32> poison, i32 1, i64 0), <vscale x 2 x i32> ... poison, <vscale x 2 x i32> zeroinitializer) ... poison, <vscale x 2 x i32> zeroinitializer) %51 = tail call <vscale x 2 x i32> @llvm.masked.load.nxv2i32.p0(ptr %47, i32 %30 = add < vscale x 2 x i32 > %29, %27... 8, <vscale x 2 x i1> %15, <vscale x 2 x i32> undef) %31 = tail call < vscale x 2 x i32 >%52 = shl <vscale x 2 x i32> %51, shufflevector (<vscale x 2 x i32> ... @llvm.aarch64.sve.ld1.gather.index.nxv2i32(<vscale x 2 x i1> %7, ptr %b, ...  $\langle vscale \times 2 \times i64 \rangle \%22$ ... insertelement (<vscale x 2 x i32> poison, i32 1, i64 0), <vscale x 2 x i32> ... poison, <vscale x 2 x i32> zeroinitializer) %32 = mul < vscale x 2 x i32 > %26, shufflevector (< vscale x 2 x i32 > %26) %53 = add < vscale x 2 x i32 > %52, %50... insertelement (<vscale x 2 x i32> poison, i32 -4, i64 0), <vscale x 2 x i32> %54 = tail call <vscale x 2 x i32> @llvm.masked.load.nxv2i32.p0(ptr %48, i32 ... poison, <vscale x 2 x i32> zeroinitializer) %33 = add < vscale x 2 x i32 > %31, %32... 8, <vscale x 2 x i1> %15, <vscale x 2 x i32> undef) %55 = mul <vscale x 2 x i32> %49, shufflevector (<vscale x 2 x i32> %34 = sdiv < vscale x 2 x i32 > %30, %33tail call void @llvm.aarch64.sve.st1.scatter.index.nxv2i32(<vscale x 2 x ... insertelement ( $\langle vscale \times 2 \times i32 \rangle$  poison, i32 -4, i64 0),  $\langle vscale \times 2 \times i32 \rangle$ ... poison, <vscale x 2 x i32> zeroinitializer) %56 = add <vscale x 2 x i32> %54, %55 ... i32> %34, <vscale x 2 x i1> %7, ptr %a, <vscale x 2 x i64> %22) %35 = mul <vscale x 2 x i32> %31, shufflevector (<vscale x 2 x i32> ... insertelement (<vscale x 2 x i32> poison, i32 5, i64 0), <vscale x 2 x i32> %57 = sdiv <vscale x 2 x i32> %53, %56 tail call void @llvm.masked.store.nxv2i32.p0(<vscale x 2 x i32> %57, ptr ... poison, <vscale x 2 x i32> zeroinitializer) .. %46, i32 8, <vscale x 2 x i1> %15) %36 = mul < vscale x 2 x i32 > %35, %34%58 = mul < vscale x 2 x i32 > %54, shufflevector (< vscale x 2 x i32 > %54) tail call void @llvm.aarch64.sve.st1.scatter.index.nxv2i32(<vscale x 2 x ... insertelement (<vscale x 2 x i32> poison, i32 5, i64 0), <vscale x 2 x i32> ... i32> %36, <vscale x 2 x i1> %7, ptr %b, <vscale x 2 x i64> %22) ... poison, <vscale x 2 x i32> zeroinitializer) %37 = shl < vscale x 2 x i32 > %36, shufflevector (< vscale x 2 x i32 >%59 = mul < vscale x 2 x i32 > %58, %57... insertelement (<vscale x 2 x i32> poison, i32 1, i64 0), <vscale x 2 x i32> tail call void @llvm.masked.store.nxv2i32.p0(<vscale x 2 x i32> %59, ptr ... poison, <vscale x 2 x i32> zeroinitializer) %38 = mul < vscale x 2 x i32 > %34, shufflevector (< vscale x 2 x i32 > %34) ... %48, i32 8, <vscale x 2 x i1> %15) %60 = shl < vscale x 2 x i32 > %59, shufflevector (< vscale x 2 x i32 > %59) ... insertelement (<vscale x 2 x i32> poison, i32 -3, i64 0), <vscale x 2 x i32> ... insertelement (<vscale x 2 x i32> poison, i32 1, i64 0), <vscale x 2 x i32> ... poison, <vscale x 2 x i32> zeroinitializer) ... poison, <vscale x 2 x i32> zeroinitializer) %39 = add < vscale x 2 x i32 > %37, %38%61 = mul < vscale x 2 x i32 > %57, shufflevector (< vscale x 2 x i32 > %57) tail call void @llvm.aarch64.sve.st1.scatter.index.nxv2i32(<vscale x 2 x ... insertelement ( $\langle vscale \times 2 \times i32 \rangle$  poison, i32 -3, i64 0),  $\langle vscale \times 2 \times i32 \rangle$ ... i32> %39, <vscale x 2 x i1> %7, ptr %c, <vscale x 2 x i64> %22) ... poison,  $\langle vscale \times 2 \times i32 \rangle$  zeroinitializer)  $\%62 = add \langle vscale \times 2 \times i32 \rangle \%60, \%61$ %40 = add i64 % index, %1%41 = tail call < vscale x 2 x i64 > @llvm.aarch64.sve.index.nxv2i64(i64 %40,tail call void @llvm.masked.store.nxv2i32.p0(<vscale x 2 x i32> %62, ptr ... i64 1) ... %47, i32 8, <vscale x 2 x i1> %15) %42 = getelementptr inbounds i8, ptr %cond, i64 %40, !dbg !43 %43 = load < vscale x 2 x i8 > , ptr %42, align 2br label %new.latch %44 = icmp ne < vscale x 2 x i8 > %43, zeroinitializer%45 = tail call i64 @llvm.aarch64.sve.cntp.nxv2i1(<vscale x 2 x i1> %44, .. <vscale x 2 x i1> %44) br label %new.latch new.latch: %63 = phi i64 [ %index, %linearized ], [ %index, %lane.gather ], [ %40, ... %uniform.block ] %64 = phi <vscale x 2 x i64> [ %uniform.vector, %linearized ], [ %22, .. %lane.gather ], [ %41, %uniform.block ] %65 = phi <vscale x 2 x i1> [ %uniform.vector.predicates, %linearized ], [ ... %23, %lane.gather ], [ %44, %uniform.block ] %66 = phi i64 [ %uniform.vec.actives, %linearized ], [ %24, %lane.gather ], ... [ %45, %uniform.block ] %67 = add i64 %63, %1%.not3 = icmp ult i64 %67, %total.iterations.to.be.vectorized br i1 %.not3, label %alc.header, label %middel.block middel.block: %condition1 = icmp eq i64 %9, 0 %68 = tail call < vscale x 2 x i32 >... @llvm.aarch64.sve.ld1.gather.index.nxv2i32(<vscale x 2 x i1> %65, ptr %a, ... <vscale x 2 x i64> %64) %69 = mul < vscale x 2 x i32 > %68, shufflevector (< vscale x 2 x i32 >... insertelement (<vscale x 2 x i32> poison, i32 18, i64 0), <vscale x 2 x i32> ... poison, <vscale x 2 x i32> zeroinitializer) %70 = tail call < vscale x 2 x i32 >... @llvm.aarch64.sve.ld1.gather.index.nxv2i32(<vscale x 2 x i1> %65, ptr %c, ... <vscale x 2 x i64> %64) %71 = shl < vscale x 2 x i32 > %70, shufflevector (< vscale x 2 x i32 > %70) ... insertelement (<vscale x 2 x i32> poison, i32 1, i64 0), <vscale x 2 x i32> ... poison, <vscale x 2 x i32> zeroinitializer) %72 = add < vscale x 2 x i32 > %71, %69%73 = tail call < vscale x 2 x i32 >... @llvm.aarch64.sve.ld1.gather.index.nxv2i32(<vscale x 2 x i1> %65, ptr %b, ... <vscale x 2 x i64> %64) %74 = mul < vscale x 2 x i32 > %68, shufflevector (< vscale x 2 x i32 >... insertelement (<vscale x 2 x i32> poison, i32 -4, i64 0), <vscale x 2 x i32> ... poison, <vscale x 2 x i32> zeroinitializer) %75 = add <vscale x 2 x i32> %73, %74 %76 = sdiv < vscale x 2 x i32 > %72, %75tail call void @llvm.aarch64.sve.st1.scatter.index.nxv2i32(<vscale x 2 x ... i32> %76, <vscale x 2 x i1> %65, ptr %a, <vscale x 2 x i64> %64) %77 = mul <vscale x 2 x i32> %73, shufflevector (<vscale x 2 x i32> ... insertelement (<vscale x 2 x i32> poison, i32 5, i64 0), <vscale x 2 x i32> ... poison, <vscale x 2 x i32> zeroinitializer) %78 = mul < vscale x 2 x i32 > %77, %76tail call void @llvm.aarch64.sve.st1.scatter.index.nxv2i32(<vscale x 2 x ... i32> %78, <vscale x 2 x i1> %65, ptr %b, <vscale x 2 x i64> %64) %79 = shl <vscale x 2 x i32> %78, shufflevector (<vscale x 2 x i32> ... insertelement (<vscale x 2 x i32> poison, i32 1, i64 0), <vscale x 2 x i32> ... poison, <vscale x 2 x i32> zeroinitializer) %80 = mul < vscale x 2 x i32 > %76, shufflevector (< vscale x 2 x i32 >... insertelement (<vscale x 2 x i32> poison, i32 -3, i64 0), <vscale x 2 x i32> ... poison, <vscale x 2 x i32> zeroinitializer) %81 = add < vscale x 2 x i32 > %79, %80tail call void @llvm.aarch64.sve.st1.scatter.index.nxv2i32(<vscale x 2 x ... i32> %81, <vscale x 2 x i1> %65, ptr %c, <vscale x 2 x i64> %64) br i1 %condition1, label %for.cond.cleanup, label %for.body.preheader12 for.body.preheader12: %indvars.iv.ph = phi i64 [ %67, %middel.block ], [ 0, %for.body.preheader ] br label %for.body, !dbg !38 for.body: %indvars.iv = phi i64 [ %indvars.iv.next, %for.inc ], [ %indvars.iv.ph, ... %for.body.preheader12 ] call void @llvm.dbg.value(metadata i64 %indvars.iv, metadata !29, metadata .. !DIExpression()), !dbg !35 %arrayidx = getelementptr inbounds i8, ptr %cond, i64 %indvars.iv, !dbg !43 %3 = load i8, ptr %arrayidx, align 1, !dbg !43, !tbaa !46, !range !50 %tobool.not = icmp eq i8 %3, 0, !dbg !43 br i1 %tobool.not, label %for.inc, label %if.then, !dbg !51 %arrayidx2 = getelementptr inbounds i32, ptr %a, i64 %indvars.iv, !dbg !52 %4 = load i32, ptr %arrayidx2, align 4, !dbg !52, !tbaa !54 %mul = mul nsw i32 %4, 18, !dbg !56 %arrayidx4 = getelementptr inbounds i32, ptr %c, i64 %indvars.iv, !dbg !57 %5 = load i32, ptr %arrayidx4, align 4, !dbg !57, !tbaa !54 %mul5 = shl nsw i32 %5, 1, !dbg !58 %add = add nsw i32 %mul5, %mul, !dbg !59 %arrayidx7 = getelementptr inbounds i32, ptr %b, i64 %indvars.iv, !dbg !60 %6 = load i32, ptr %arrayidx7, align 4, !dbg !60, !tbaa !54 %mul10.neg = mul i32 %4, -4, !dbg !61 %sub = add i32 %6, %mul10.neg, !dbg !62 %div = sdiv i32 %add, %sub, !dbg !63 store i32 %div, ptr %arrayidx2, align 4, !dbg !64, !tbaa !54 %mul15 = mul nsw i32 %6, 5, !dbg !65 %mul18 = mul nsw i32 %mul15, %div, !dbg !66 store i32 %mul18, ptr %arrayidx7, align 4, !dbg !67, !tbaa !54 %mul23 = shl nsw i32 %mul18, 1, !dbg !68 %mul26.neg = mul i32 %div, -3, !dbg !69 %sub27 = add i32 %mul23, %mul26.neg, !dbg !70 store i32 %sub27, ptr %arrayidx4, align 4, !dbg !71, !tbaa !54 br label %for.inc, !dbg !72 for.inc: %indvars.iv.next = add nuw nsw i64 %indvars.iv, 1, !dbg !73 call void @llvm.dbg.value(metadata i64 %indvars.iv.next, metadata !29, ... metadata !DIExpression()), !dbg !35 %exitcond.not = icmp eq i64 %indvars.iv.next. %wide.trip.count. !dbq !36 br i1 %exitcond.not, label %for.cond.cleanup, label %for.body, !dbg !38, ...!llvm.loop!74 for.cond.cleanup: tail call void asm sideeffect ".inst 0x2520e040", ""() #7, !dbg !39, !srcloc ... !41

> ret void, !dbg !42 CFG for 'foo' function